Applying a Controlled Medical Terminology to a Distributed, Production Clinical Information System

Bruce H. Forman, M.D., James J. Cimino, M.D.,
Stephen B. Johnson, Ph.D., Soumitra Sengupta, Ph.D.,
Robert Sideli, M.D., Paul Clayton, Ph.D.
Department of Medical Informatics
Columbia University
Columbia-Presbyterian Medical Center
New York, NY

To maximize the value of computerized medical records systems, an organizing structure is needed. That structure can be provided by a controlled medical terminology (CMT). At Columbia-Presbyterian Medical Center, we have been employing a controlled medical terminology, our Medical Entities Dictionary (MED), to mediate the storage and retrieval of patient data and enable decision support applications. This paper describes how the MED is actually used for data management in our distributed clinical information systems environment. Our system tools which access the MED for production purposes facilitate the mapping of terms from many sources to a uniform representation of concepts and also return information about the relationships between concepts. Applications which access a CMT for production purposes should be optimized for performance in high volume settings, fault tolerant, synchronizable, extensible, portable, and maintainable. We briefly describe our system architecture and then demonstrate how we utilize the MED for translation and semantic information as data is moved into and out of our patient database. We discuss our current tools and present a preview of the next generation of applications which will manage access to the MED for our production systems.

INTRODUCTION

In order for a computer-based medical record system to be more than just an electronic collection of unrelated data, an underlying structure is required [1]. A level of organization is necessary to ensure the efficiency of data storage and retrieval, achieve functional integration between systems, enhance consistency and accuracy of information, and enable decision support [2]. The useful exchange of information between applications and institutions requires a common ground for framing the structure and content of the medical record [3]. Applying an organizing framework to data transforms it into useful information which is analyzable and comparable.

A controlled medical terminology (CMT) has been recognized as the structure around which medical information systems should be organized [4]. Indeed, the majority of the most well-developed medical information systems maintain a CMT to facilitate efficient storage and display of information and enable decision support [5-8].

At Columbia-Presbyterian Medical Center (CPMC), we have been applying a CMT, our Medical Entities Dictionary (MED), in order to store and retrieve data to and from our central patient repository and structure medical logic modules for decision support applications [5, 9]. Because we have multiple departmental applications running on a number of platforms which are interfaced to our patient repository using several different paths, we have developed a distributed methodology for applying and managing the use of the MED in our production environment. While the MED has been described in the literature and its role mentioned in reports of clinical systems [10, 11], there is no published description of how it is used to manage clinical data.

This paper reviews the kinds of information needed from a CMT working in conjunction with a clinical information system (CIS) for the structured storage and retrieval of data. We delineate desirable characteristics of applications which manage the use of a CMT in a production environment. After a brief description of our systems architecture, we describe the actual methods used and results achieved in applying our MED to a selected production system. We conclude with a discussion of lessons learned and our future directions in this area.

REQUIREMENTS

Caregivers require a view of patient data organized along clinical lines. Thus, for example, a physician may want to see the results of all recent blood glucose

0195-4210/95/$5.00

tests even though those tests may have been performed by distinct lab systems which code the tests differently [12]. Thus, one requirement of an integrated clinical information system is that it has the facility to consistently resolve disparate representations of the same medical concept. The entity which maps semantically equivalent terms to a single concept is a CMT [4].

A second requirement of clinical information systems is the ability to retrieve and filter information by class. For example, a radiology information system considers an AP Chest Xray exam and a PA and Lateral Chest Xray series as two distinct entities because their billing characteristics differ. However, clinicians interested in viewing the latest results of thoracic imaging studies or clinical researchers attempting to gather reports of all chest xrays would consider these two studies to be equivalent for their immediate data needs. Thus, CIS's need to be able to represent and retrieve related data. Information about such associations is properly held and managed by a CMT because its structure contains links between related terms [2].

Therefore, system tools utilizing a CMT must facilitate the mapping of terms from many sources to a uniform representation of concepts. Such tools should also be able to return information about the relationships between concepts such as class membership.

DESIRABLE CHARACTERISTICS OF TOOLS WHICH UTILIZE CMT'S

System tools which utilize CMT's for performing term translations or obtaining relationship information between terms in distributed production environments should have certain characteristics. These applications should be optimized for performance in high volume settings, fault tolerant, synchronizable, extensible, portable, and maintainable.

If tools which access a CMT operate on a large volume of data moving into or out of a patient database, their performance needs to be optimized. They cannot significantly hinder processing whether it is for the display of results or the storage of critical data.

If accessing a CMT becomes a critical step in either encoding or decoding data as it moves between a repository and client applications, then the tools which mediate this access must be fault tolerant and should be able to continue functioning under certain adverse conditions. For example, although centralized storage of a CMT facilitates its maintenance and ensures consistency across an enterprise, caching of translation and hierarchy data on ancillary processors can insulate these systems from temporary degradation in network throughput or even network outages.

In a distributed environment, many different systems will need access to a CMT. If applications which utilize a CMT are designed to maintain local copies of CMT subsets for performance reasons, then a method of synchronizing these multiple versions is obligatory.

A CMT will grow continuously, reflecting the ever-changing nature of medical knowledge. Thus, the tools which mediate access to a CMT for production purposes must be designed to reflect changes in the CMT as soon as they occur. Lastly, these applications should be adaptable to many operating environments and be easily maintainable. For example, they should be modular and should not require special knowledge of the underlying CMT they manipulate.

CPMC ARCHITECTURE

The components of the CPMC systems architecture have been described elsewhere [5, 13-16]. Briefly, the CPMC architecture can be described as a series of concentric logical layers including (from outermost to innermost): message handling, encoding, message routing, monitoring, data access, and data storage.

Most relevant to this paper are the encoding and data storage layers. Translation of data from a local coding scheme to that used by our MED or visa versa is done in the encoding layer.

Functionally, the MED, as described by Cimino, et. al. [5], uses a semantic network model that includes a classification hierarchy. Each concept in the MED is assigned a unique integer identifier. The parent-child relationships between nodes on the network correspond to the classification of the concepts. The concepts are characterized by named attributes or slots which may or may not have values.

The MED provides several distinct advantages for clinical computing support [5]. It provides a central resource containing current terms in ancillary systems. Its ability to represent semantic information simplifies the design of our clinical database. The MED shields application and clinical decision support developers from having to synchronize their programs with changes in ancillary systems. Lastly, the knowledge in the MED facilitates its own maintenance.

The MED utilized in our production environment is maintained in shared memory on a UNIX* workstation which functions as one of our interface engines and in memory-resident relational tables on our mainframe. Because a number of applications need to access MED information for real-time processing, a memory resident architecture was chosen to optimize efficiency.

The storage layer corresponds to our patient data repository which is a heterogeneous database currently distributed among DB2 (relational), VSAM (indexed), and IMS (hierarchical) files on a mainframe. Lab, radiology, pharmacy, demographic, outpatient, and resident signout data are currently being stored in DB2. Going forward, our intent is to store all patient data on DB2.

METHODS FOR APPLYING THE MED

The MED is used extensively to mediate the movement of data between production systems at CPMC. It is the unifying entity that permits structured data storage, retrieval and analysis. The MED is currently used to encode data for storage in our patient database from three of our highest volume ancillary systems: clinical laboratory, radiology, and pharmacy. We will illustrate the use of the MED in conjunction with our clinical lab system.

Data Routing and Storage

Vendor codes for lab exams are transformed into MED codes during the passage of result data through an interface on the way to storage on our mainframe. This translation is mediated by extract programs which generate subsets of the MED containing vendor codes, stored as slot values (attributes) of particular concepts in the MED and associated with MED codes. When a MED-mediated translation process is needed for a new set of data, a new extract program is created by a developer based on the sets of MED classes required for the encoding process. Subsequent changes to a MED class covered by an extract program will be reflected in the MED subset required for translation when the extract is rerun. Extract programs are rerun and the resultant updated MED subset tables are propagated to appropriate systems automatically every time the MED is altered.

We require more than just translation information from the MED. We must also take advantage of semantic information such as class membership. For example, we need to filter microbiology result messages so that we can transmit copies of positive reports to an infection tracking database system. Phrases used by the microbiology lab when interpreting a culture are encoded according to the MED during the process of uploading the result to our CIS. The MED codes corresponding to these result messages are members of one of two MED classes: "Positive Culture Result" or "Negative Culture Result." Determining whether a microbiology message should be sent to our infection tracking database, therefore, is simply a matter of determining whether the MED code for the culture result is a member of the "Positive Culture Result" class.

Data Retrieval and Display

When users request the display of clinical data from our patient database, their request accesses display programs which decode data according to the MED so that it is easily readable. This translation is mediated by the relational structure which contains the MED on our mainframe. These memory-resident tables are updated automatically each time the MED is altered. This update is synchronized with that performed on the MED extracts.

Data Queries and Decision Support

We are beginning to use periodic data queries to populate departmental databases with lab results and to return patient results to intelligent end-user workstations. These applications perform their lookups according to MED classes. Our medical logic modules (MLM) which drive the generation of clinical alerts and reminders are also keyed to MED codes.

RESULTS

For illustrative purposes, we again focus on our lab system. We have been using the MED to mediate the upload of lab results since 1990. We currently upload an average of 9,400 lab tests per day during the working week to our patient repository. Each of these reports has key fields translated into MED codes during passage through an interface engine. Each day, an average of 18,000 requests for lab data are received from end-users by our CIS.

We experience an average error rate in translating between a vendor coding scheme and MED codes of 0.22 errors per day for lab results. Translation errors are logged by our interface engine and emailed to system managers. Almost all of these errors are due to vendor codes which have not yet been associated with MED concepts.

Requests for MED updates are currently directed to one of two faculty members in our department. A total

*UNIX is a registered trademark.

of 91 MED updates have been performed in the last 20 months for an average of 4.55 updates per month. Updates consist of a log file with one line of text per MED transaction. Valid transactions include adding MED codes and populating or updating slot values for a MED code. For example, the most recent MED update file as of this writing contained 3074 lines, including the addition of 83 new concepts and 1627 slot values.

DISCUSSION

We continue to add new sources of data to our patient repository. In order to maximize the utility of our integrated patient database, we intend to model the vocabularies of the domains represented by these sources in the MED, encode results from these sources according to the MED for storage, and utilize this structured data for decision support. Thus, the applications we use to access MED information have become critical components of our architecture.

The increasing need to access the MED for production applications has highlighted areas for potential improvement in our existing tools. Our current methodology of maintaining an extract of the MED for translation purposes has the advantages of being fault tolerant, efficient, and complete for a given domain.

Our extract tables are propagated to and reside on platforms where local translation processes occur. Thus, central server access is not required for each translation operation and temporary network outages will not disrupt the translations. However, a new extract must be designed for each domain in which translations are needed. This requires not only application development but also foreknowledge of all possible classes of MED codes needed for translations. Major changes in an ancillary system can require a change in the program that generates the extract, if, for example, an entirely new class of MED codes is needed for a translation. In addition, although the extracts represent small subsets of the overall MED, only a portion of the extracts themselves are ever used with any frequency. Lastly, the extracts are generally designed to fulfill a circumscribed purpose. Thus, they are now designed to perform only translations. We resort to hard-coded tables for those applications which need to garner hierarchy information from the MED such as class membership.

The shortcomings listed above have motivated us to begin development of the next generation of MED access tools. We are currently testing a more generalized application set which queries the shared memory version of our MED for information via a library of remote procedure calls (RPC's) which run in a TCP/IP environment.

The ability to query the MED for information with a library of RPC's obviates the necessity of knowing *a priori* what MED classes and codes will be needed for a given translation. Also, the addition of a new class of information to the MED for a particular domain does not require that we alter and re-propagate our RPC tools the way we have to with our MED extracts.

We chose to use an RPC library so that we could access MED information from any TCP/IP capable host on our network. This obviates the necessity of duplicating and maintaining the MED's shared memory structure on multiple platforms.

The new methodology also has the advantage of being more flexible in terms of the kinds of information that can be returned. Using appropriate calls, we can retrieve translation or hierarchy information with equal facility.

The greatest challenge we face as we develop this new methodology to access the MED is to maintain the processing efficiency and fault tolerance of our existing tools. A centralized vocabulary server providing on-line, real-time service to a series of remote clients and the communications links upon which it depends must be highly reliable and efficient [17] since they will be on the critical paths of multiple key clinical upload processes.

The critical importance of the timely movement of clinical data from ancillary systems to our patient database has motivated us to strike a balance between generalized remote access to a centrally maintained vocabulary facility and the efficiency and safety of terminology data held locally where it needs to be used. Thus, we are currently testing the utility of maintaining caches of MED information locally on a given platform, which will be populated and refreshed by using the RPC library.

CONCLUSIONS

An essential component of an integrated electronic medical record system is a CMT. A CMT permits structured data storage and retrieval enabling accurate and efficient data access. It normalizes, disambiguates, and eliminates redundancies between coding schemes within an enterprise and between enterprises permitting aggregate data analysis. Lastly, a CMT provides the structure which enables diagnostic decision support. With the proper tools, a CMT can add value to a production CIS environment.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions of programmers Nilesh Desai, Socrates Socratous, and Taejin Yoon. This publication was developed under the auspices of the Columbia University Center for Advanced Technology (CAT) in High Performance Computing and Communications in Health Care, a New York State CAT supported by the New York State Science and Technology Foundation. MED development was supported in part by the IBM Corporation.

References

[1] Rector AL, Nowlan WA, Kay S. Foundations for an Electronic Medical Record. Methods of Information in Medicine. 1991; 30(3) 179-86.

[2] Linnarsson R, Wigertz O. The Data Dictionary - A Controlled Vocabulary for Integrating Clinical Databases and Medical Knowledge Bases. Methods of Information in Medicine. 1989; 28(2) 78-85.

[3] Board of Directors of the American Medical Informatics Association. Standards for Medical Identifiers, Codes, and Messages Needed to Create an Efficient Computer-stored Medical Record. JAMIA. 1994; 1(1): 1-7.

[4] Huff SM, Craig RB, Gould BL, Castagno DL, Smilan RE. Medical Data Dictionary for Decision Support Applications. In: Stead WW, editor. Proceedings of the 11th SCAMC. 1987; 310-317.

[5] Cimino JJ, Clayton PD, Hripcsak G, Johnson SB. Knowledge-Based Approaches to the Maintenance of a Large Controlled Medical Terminology. JAMIA. 1994; 1(1): 35-50.

[6] Pryor TA, Gardner RM, Clayton PD, Warner HR. The HELP System. Journal of Medical Systems. 1983; 7(2): 87-102.

[7] Stead WW, Hammond WE. Computer-Based Medical Records: The Centerpiece of TMR. MD Computing. 1988; 5(5): 48-62.

[8] McDonald CJ, Blevins L, Tierney WM, Martin DK. The Regenstrief Medical Records. MD Computing. 1988; 5(5): 34-47.

[9] Cimino JJ, Hripcsak G, Johnson SB, Clayton PD. Designing an Introspective, Multi-Purpose Controlled Medical Vocabulary. In: Kingsland LC, editor. Proceedings of the 13th SCAMC. 1989; 513-518.

[10] Cimino JJ, Clayton PD. Coping with Changing Controlled Vocabularies. In: Ozbolt JG, editor. Proceedings of the 18th SCAMC. 1994; 135-9.

[11] Barrows RC, Cimino JJ, Clayton PD. Mapping Clinically Useful Terminology to a Controlled Medical Vocabulary. In: Ozbolt JG, editor. Proceedings of the 18th SCAMC. 1994; 211-5.

[12] McDonald CJ, Tierney WM, Overhage JM, Martin DK, Wilson DK. The Regenstrief Medical Record System: 20 Years of Experience in Hospitals, Clinics, and Neighborhood Health Centers. MD Computing. 1992; 9(4): 206-17.

[13] Friedman C, Hripcsak G, Johnson SB, Cimino JJ, Clayton PD. A Generalized Relational Schema for an Integrated Clinical Patient Database. In: Miller RA, editor. Proceedings of the 14th SCAMC. 1990; 335-339.

[14] Hripcsak G, Cimino JJ, Johnson SB, Clayton PD. The Columbia-Presbyterian Medical Center Decision-Support System as a Model for Implementing the Arden Syntax. In: Clayton PD, editor. Proceedings of the 15th SCAMC. 1991; 248-252.

[15] Clayton PD, Sideli RV, Sengupta S. Open Architecture and Integrated Information at Columbia-Presbyterian Medical Center. MD Computing. 1992; 9(5): 297-303.

[16] Johnson SB, Forman B, Sengupta S, Sideli R, Cimino J, Clayton, P. The Electronic Medical Record: Architecture and Standards. Proceedings, "Toward An Electronic Patient Record '95," Orlando FL; Medical Records Institute; 1995; 2: 14-18.

[17] Rocha RA, Huff SM, Haug PJ, Warner HR. Designing a Controlled Medical Vocabulary Server: The VOSER Project. Computers and Biomedical Research. 1994; 27(6): 472-507.